SOCRATIC Pretraining: Question-Driven Pretraining for Controllable Summarization

Artidoro Pagnoni[*1] **Alexander R. Fabbri**[2] **Wojciech Kryściński**[2] **Chien-Sheng Wu**[2]
[1]University of Washington, [2]Salesforce AI Research
artidoro@uw.edu, {afabbri, wojciech.kryscinski, wu.jason}@salesforce.com

Abstract

In long document controllable summarization, where labeled data is scarce, pretrained models struggle to adapt to the task and effectively respond to user queries. In this paper, we introduce SOCRATIC pretraining, a question-driven, unsupervised pretraining objective specifically designed to improve controllability in summarization tasks. By training a model to generate and answer relevant questions in a given context, SOCRATIC pretraining enables the model to more effectively adhere to user-provided queries and identify relevant content to be summarized. We demonstrate the effectiveness of this approach through extensive experimentation on two summarization domains, short stories and dialogue, and multiple control strategies: keywords, questions, and factoid QA pairs. Our pretraining method relies only on unlabeled documents and a question generation system and outperforms pre-finetuning approaches that use additional supervised data. Furthermore, our results show that SOCRATIC pretraining cuts task-specific labeled data requirements in half, is more faithful to user-provided queries, and achieves state-of-the-art performance on QMSum and SQuALITY.

1 Introduction

Summarization systems are designed to help users navigate large amounts of information (Edmunds and Morris, 2000), but often fail to meet the unique needs of different users, especially for long documents. Recent research has explored ways to make summarization systems more controllable (Bornstein et al., 1999; Leuski et al., 2003) by allowing users to input queries or control sequences such as keywords (He et al., 2020), questions (Zhong et al., 2021), entity chains (Narayan et al., 2021), or question-answer pairs (Narayan et al., 2022).

A challenge shared by all of the mentioned approaches is the absence of abundant labeled data.

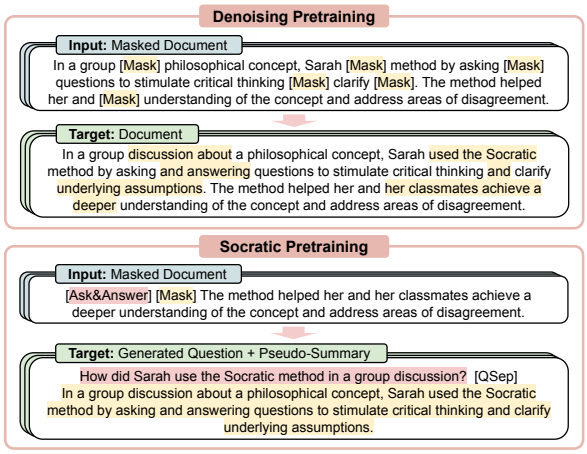


Figure 1: Our SOCRATIC pretraining compared to denoising. We mask important sentences in unlabeled input documents and train the model to generate both **questions** and **pseudo-summaries** as their answers.

Currently available datasets for training these systems are the result of expensive annotation efforts (Zhong et al., 2021; Kulkarni et al., 2020; Wang et al., 2022) with only hundreds to a few thousand query-document pairs, with the same document often being repeated. This translates into poor adherence of generated summaries to user-provided queries, particularly when these are finegrained plans. Recent work demonstrates the benefits of tailoring the pretraining objective to downstream task characteristics, especially where training data is difficult to obtain in large quantities like factuality-focused and multi-document summarization (Wan and Bansal, 2022; Xiao et al., 2022). In controllable summarization, summaries are grounded by queries, so designing an objective for the task requires introducing realistic queries in unlabeled data in a scalable manner.

This work introduces SOCRATIC pretraining, an unsupervised pretraining objective for language models that is specifically designed for controllable summarization. It is inspired by the Socratic method and aims to facilitate the identification of

[*] Work done during internship at Salesforce

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12737–12755
July 9-14, 2023

relevant content and ensure that the generated summary faithfully responds to the user query. During SOCRATIC pretraining (see Figure 1) the language model is trained to generate relevant questions based on an input document and then answer them, bringing finegrained controllability to model pretraining which translates to better adherence to user queries.

SOCRATIC pretraining only relies on unlabeled data and a question generation system and outperforms pre-finetuning approaches relying on additional supervised data (Aghajanyan et al., 2021; Wei et al., 2021; Fabbri et al., 2021a). In this work, we demonstrate the effectiveness of the SOCRATIC objective through *pretraining adaptation*, where a language model is further pretrained with the SOCRATIC objective before finetuning on task-specific labeled data.

In summary, our contributions are as follows[1]:

- We introduce the SOCRATIC pretraining objective for controllable summarization to improve adherence to user-specified queries or plans, both high-level and finegrained.
- We show that SOCRATIC pretraining performs well across domains, control strategies, and achieves state-of-the-art performance on two datasets.
- We perform ablations on our approach showing that SOCRATIC pretraining cuts labeled data requirements in half.

2 Related Work

Task-Specific Pretraining Adaptation Current state-of-the-art methods in abstractive summarization apply a two-step approach where models are first pretrained on large corpora of text with task-agnostic variations of the text denoising objective and next finetuned on labeled examples from the target task (Lewis et al., 2020; Raffel et al., 2020).

However, in tasks where labeled data is scarce, task-specific pretraining objectives have been shown to provide significant benefits. Recent work adapted language models to summarize multiple documents (Xiao et al., 2022), produce more factual summaries (Wan and Bansal, 2022), or plan with entity chains (Narayan et al., 2021). We build on these methods, focusing on the downstream task of controllable summarization.

[1]Our code is be available at https://github.com/salesforce/socratic-pretraining

Other studies demonstrate the effect of continued pretraining (Gururangan et al., 2020) and pre-finetuning (Aghajanyan et al., 2021; Wei et al., 2021; Fabbri et al., 2021a) on downstream task adaptation. These either continue training with the same objective on data in the downstream task domain or perform multitask learning using labeled data. In this work, we demonstrate the benefits of language model adaptation with a task-specific pretraining objective without additional supervised data and show that these benefits are consistent and statistically significant in low-resource settings like query-focused summarization (QFS).

Controllable Summarization Controllable text generation (Hu et al., 2017) aims to control properties of the generated text including style (Kumar et al., 2021), length, or content (Fan et al., 2018; He et al., 2020). Approaches for content control vary according to the type of control: keywords (He et al., 2020), entities (Narayan et al., 2021), questions (Vig et al., 2022), factoid question-answer pairs (also called QA blueprints) (Narayan et al., 2022). As opposed to methods like GSum (Dou et al., 2021), which insert control tokens on the encoder side, we focus on decoder-based methods which do not require re-encoding the document when the control sequences are updated. In summarization, these controls can broadly indicate the information to summarize, like the questions in query-focused summarization, or provide a detailed plan of the text to be generated, like the entity chains. While these types of control are not typically studied together we show that our SOCRATIC pretraining provides benefits across the board for both high-level and finegrained queries and plans.

Learning with Questions Inspired by the Socratic method, recent literature in education theory shows students generate questions as a way of learning (Rosenshine et al., 1996; Aflalo, 2021), hinting at the potential benefits that could derive from incorporating questions during model training. Previous work shows that question-answer pairs, both generated (Du et al., 2017; Alberti et al., 2019; Ko et al., 2021; Murakhovs'ka et al., 2022; Chakrabarty et al., 2022) and from the web (Narayan et al., 2020), can provide useful training signal for pretrained encoders (Jia et al., 2022) as well as question generation and abstractive summarization systems (Narayan et al., 2022). Our SOCRATIC objective builds on these observations

and is designed to improve sequence-to-sequence model pretraining for more controllable summarization systems. Similar to information-seeking Dialogue Inpainting (Dai et al., 2022), SOCRATIC pretraining extracts questions from unlabeled data focusing on higher-level questions, whose answers are full sentences, instead of factoid QA pairs.

3 SOCRATIC Pretraining

During SOCRATIC pretraining, the model takes as input a document with important sentences masked and is trained to generate questions about the masked content and produce the mask itself. As seen in Figure 1, SOCRATIC pretraining is formulated as a sequence-to-sequence task and consists of two steps 1) important content is selected from unlabeled documents to be masked, and 2) a question-generation system is applied to produce questions about the selected content. The question augmentation component trains the model to produce summaries grounded to questions and allows for controllability as the end-user can prompt the model decoder with new questions during inference. We describe both steps below.

3.1 Content Selection

Selecting important content is essential for the model to learn to generate salient questions and summaries. In SOCRATIC pretraining, this content selection is done using the PEGASUS-style Gap Sentence Generation (GSG) objective (Zhang et al., 2020a), which we now briefly describe. Sentences with the highest self-Rouge with the document are selected for masking, ensuring that there is high information overlap with the rest of the document. The selected sentences, concatenated, produce a *pseudo-summary* of the document. As in PEGASUS, a Gap Sentence Ratio (GSR) of 45% is used, meaning that 45% of the sentences in the document are selected to appear in the target pseudo-summary. To help the model learn to copy, 80% of these sentences are masked and 20% are kept unmasked in the input document. Documents and summaries are truncated to 512 and 256 tokens.

3.2 Question Augmentation

After selecting the pseudo-summary, a question generation (QG) system is applied to obtain a question from each sentence of the pseudo-summary. The QG system takes as input one of the selected sentences at a time and the unmasked document as context. We apply MixQG (Murakhovs'ka et al., 2022), a state-of-the-art QG system.

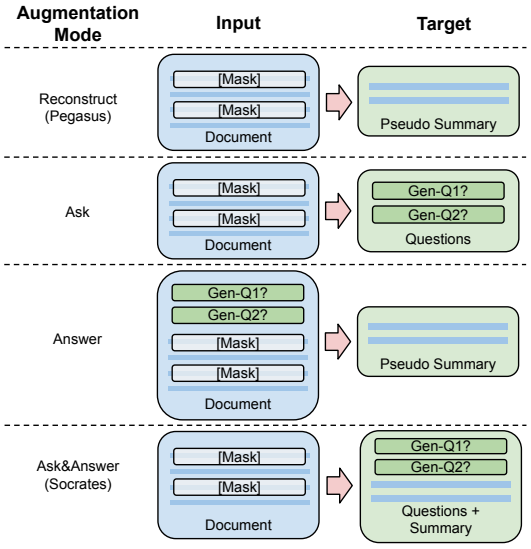


Figure 2: SOCRATIC augmentation modes vs. Pegasus.

The choice to generate a question for each selected sentence, as opposed to each entity or the entire summary, is driven by three reasons. First, sentences in the pseudo-summary are selected from across the document and generally lack coherence, so there is no single query they collectively answer. Second, current QG systems are not trained to produce paragraph-level questions. Third, entity-level questions are often simple paraphrases of the answer sentence and are uncommon in QFS datasets.

Questions whose answers are full sentences, therefore, offer a compromise in terms of the complexity of the question and the coherence of the answer. We refer to these sentence-level questions as *content-questions* as they tend to ask about the content of the document instead of specific entities.

3.3 Training Objective

After obtaining the questions, there are multiple ways to introduce them in the training objective either in the input or in the target text. As seen in Figure 2, we experiment with three modes on top of the base GSG objective:

- *Reconstruct.* The reconstruct mode is the default GSG mode where no questions are introduced. The masked document is the input and the pseudo-summary is the target text. We provide this mode as a baseline for our approach.
- *Ask.* Given the masked document as input, the model is trained to only predict the questions

about the masked sentences. This is the only mode where the target text does not include the pseudo-summary. With this mode, the model is trained to predict which questions can be asked in a given context.

- *Answer.* Here, the questions are prepended to the masked input document while the target text remains the pseudo-summary. This mode is similar to how queries are introduced to the model during query-focused summarization and should help the model learn to respond to user-provided queries. However, this mode forgoes content planning as each generated sentence corresponds to one of the questions prepended to the input.

- *Ask&Answer.* This mode combines benefits from both *Ask* and *Answer* modes. The model is tasked to first generate questions about the document and then, conditioning on both the document and the generated questions, the pseudo-summary. The model conditions on the generated questions in the decoder. This mode can be seen as first generating a fine-grained plan for the pseudo-summary and then the pseudo-summary itself.

Like Tay et al. (2022), we prepend special tokens `<ask>`, `<answer>`, and `<ask&answer>` to the input document to specify the augmentation mode, and the `<qsep>` token to separate the generated questions from the target pseudo-summary.

4 Experimental Setup

We describe the experimental setup that we use to study SOCRATIC pretraining along with empirical studies justifying our design decisions.

4.1 Model Architecture

The SOCRATIC objective can be applied to any sequence-to-sequence language model irrespective of its specific architecture. In our experiments, we choose BART-large (Lewis et al., 2020), as the starting point for SOCRATIC pretraining adaptation. Following previous work on pretraining adaption for summarization, we pick BART over PEGASUS for its smaller size without performance compromises on summarization benchmarks and its more general-purpose pretraining objective. BART is also the underlying model in the SegEnc (Vig et al., 2022) architecture, which achieved state-of-the-art performance on QMSum, outperforming models such as LongT5 (Guo et al., 2022).

Instead of pretraining the language model from scratch, we demonstrate the effectiveness of the proposed objective through what we call *pretraining adaptation*, where a generic language model is further pretrained with the SOCRATIC objective before being finetuned on task-specific labeled data. Although we introduce a new term for this training phase, *pretraining adaptation* was recently employed to evaluate task-specific pretraining objectives for factuality and multi-document summarization (Wan and Bansal, 2022; Xiao et al., 2022).

After SOCRATIC pretraining adaptation, the resulting model is used to initialize the SegEnc architecture, which is then finetuned on labeled data from downstream tasks. Pretraining and finetuning hyperparameter details are available in A.2.

4.2 Pretraining Corpus

We experiment with three different corpora, two of which are part of the Pile (Gao et al., 2021).

- *OpenWebText2* is a web-scraped dataset inspired by WebText (Radford et al., 2019) that uses Reddit upvotes of outgoing links as a proxy for page quality. Raffel et al. (2020) found this dataset to work well for summarization pretraining.

- *Books3* is a collection of both fiction and non-fiction books. We explore this data because our downstream tasks involve the short story and dialogue domains, and Csaky and Recski (2021) show books can be a good source of dialogue data.

- *UnDial* (He et al., 2022) We also explore using a dialogue corpus. As there are only two speakers in each dialogue in UnDial, we use a simple rule-based system to convert dialogues to third person. The pseudo-summary and related questions are then expressed in the third person while the input remains in the original dialogue format.

4.3 Downstream Tasks

To determine whether SOCRATIC pretraining improves model initialization for finetuning on controllable summarization, we test on two downstream datasets for query-focused, long-document summarization: QMSum and SQuALITY (dataset statistics can be found in A.1). We focus on long

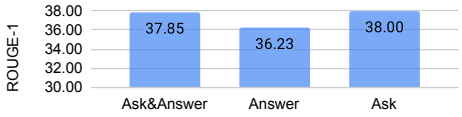


Figure 3: Comparison of question augmentation modes.

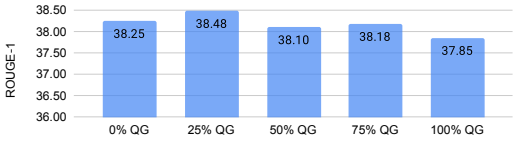


Figure 4: Comparison of QG augmentation proportions.

document datasets as a challenging and practical testbed for controllable summarization methods.

QMSum. QMSum is a benchmark for query-based, multi-domain meeting summarization (Zhong et al., 2021). The dataset consists of 1,808 query-summary pairs over 232 meetings, including product, academic, and parliamentary meetings.

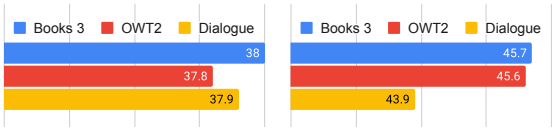
SQuALITY. SQuALITY is a dataset for query-based short stories summarization (Wang et al., 2022). The dataset is composed of 625 examples over 100 stories with four long reference summaries per document-question pair.

4.4 Evaluation Protocol

We apply the standard Rouge (Lin, 2004) and BERTScore (Zhang et al., 2020b) metrics to compare model generations with reference summaries on downstream finetuning tasks. In SQuALITY, we use the same procedure as the dataset authors to incorporate multiple references by taking the maximum score over the reference summaries. We also conduct a human evaluation study to ensure the variations between models are meaningful to users. Details on the setup can be found in A.4.

5 SOCRATIC Pretraining Ablations

In this section, we corroborate our design choices with ablation studies of the components of SOCRATIC pretraining. Similar to Zhang et al. (2020a) and Raffel et al. (2020), to save time and resources, we conduct the ablations of the objective on a small scale by restricting the pretraining adaptation to 1M documents from the OpenWebText2 corpus and then finetuning it on the full downstream task datasets. We report the mean over five randomly initialized finetuning runs on the validation set.



(a) QMSum (b) Squality

Figure 5: Effect of the pretraining corpus (dev set).

Question Augmentation Modes In Figure 3, we compare the performance of the three approaches for incorporating generated questions in the SOCRATIC objective. The *Ask* and *Ask&Answer* perform similarly while *Answer* lags behind. This is in line with our hypothesis that learning which questions are relevant in a given context is a useful training signal for the model. The *Ask&Answer* mode also grounds the pseudo-summary generation in a sequence of finegrained questions. Therefore, it is chosen to be used in SOCRATIC pretraining.

Question Augmentation Proportion Incorporating questions with the *Ask&Answer* mode in each pretraining example could bias the model to always start by generating questions. We hypothesize that combining the *Reconstruct* mode with the *Ask&Answer* mode could alleviate this bias. In Figure 4, we find that introducing questions in 25% of pretraining examples leads to the best performance and use this proportion when scaling the pretraining adaptation.

Pretraining Corpus Selection In Figure 5, we find that the choice of pretraining corpus has a small but consistent effect on the performance of the SOCRATIC pretrained model on downstream tasks. The Books3 corpus performs best both on QMSum and SQuALITY. The dialogue corpus offers a slight advantage over OpenWebText2 on QMSum, a dialogue summarization task, while the opposite is true for SQuALITY. As a result, the full Books3 corpus, consisting of 30M training instances, is used in further experiments.

6 Query Focused Summarization Results

We scale the SOCRATIC pretraining adaptation based on the findings of the previous ablation and evaluate its downstream effects on query-focused summarization. Unless specified, the results in this section are averaged over five randomly initialized finetuning runs on the downstream tasks.

In Table 1, we compare the effect of SOCRATIC pretraining to other pretraining strategies on QMSum and SQuALITY. We obtain an improvement

Model	Rouge1	Rouge2	RougeL	BS-R
QMSum				
BART-LS (Xiong et al., 2022)	37.90	12.10	33.10	-
BART-Large SegEnc	37.05	13.04	32.62	87.44
+ *WikiSum Pre-Finetuning*	*37.80*	*13.43*	*33.38*	-
+ BART Pret. 1M	36.64	12.44	31.94	86.94
+ SOCRATIC Pret. 1M	37.46	13.32	32.79	87.54
+ PEGASUS Pret.	37.29	13.30	32.70	87.48
+ SOCRATIC Pret.	**38.06**	**13.74**	**33.51**	**87.63**
Squality				
LED	27.7	5.9	17.7	-
PEGASUS	38.2	9.0	20.2	-
BART	40.2	10.4	20.8	-
BART + DPR	41.5	11.4	21.0	-
Human	46.6	12.5	22.7	-
BART-Large SegEnc	45.68	14.51	22.47	85.86
+ PEGASUS Pret.	45.78	14.43	**22.90**	85.94
+ SOCRATIC Pret.	**46.31**	**14.80**	22.76	**86.04**

Table 1: Results on QMSum and SQuALITY with pretraining on Books3. Baselines from Vig et al. (2022) and Wang et al. (2022) respectively. 1M indicates that 1M pretraining instances are used.

of +1.01 and +0.53 Rouge-1, respectively, surpassing even the use of additional supervision from the related dataset WikiSum in Vig et al. (2022) and achieving new state-of-the-art results. These improvements are validated by a human study reported in Figure 6 and showing that SOCRATIC SegEnc performs better than the baselines in 59-65% of instances. Details of the human evaluation are found in A.4.

6.1 Disentangling the Effect of Questions

The main baseline for SOCRATIC pretraining is the PEGASUS style GSG pretraining. We therefore perform a pretraining adaptation of BART-large with the GSG objective on the full Books3 corpus. In Table 1, we observe that GSG pretraining on the full Books3 corpus improves by +0.24 Rouge-1 over the BART SegEnc model. However, with the SOCRATIC objective, 1M examples from Books3 (1/30 of the full corpus) are sufficient to surpass GSG pretraining, with a +0.41 Rouge-1 improvement over BART SegEnc. This indicates that GSG pretraining, tailored to generic summarization, is only marginally helpful in tasks where summaries have to answer user-provided queries. In addition, increasing the corpus for SOCRATIC pretraining to the entire Books3 corpus further improves the performance by +0.60 Rouge-1 on QMSum, showing that the benefits of the pretraining objective do not saturate early and that the model continues to improve with additional SOCRATIC pretraining.

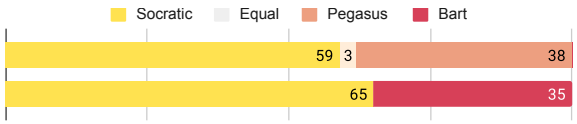


Figure 6: Human annotators' preferences on QMSum.

We also compare to BART-LS, an orthogonal approach that tailors BART's architecture, pretraining corpus, and objective to long documents (Xiong et al., 2022). While our approaches are complementary, we outperform BART-LS on QMSum by +1.64 Rouge-2. This confirms our hypothesis that grounding generations in control queries in SOCRATIC pretraining is beneficial in controllable summarization, even more so than better long document modeling.

6.2 Comparing to Continued Pretraining

Gururangan et al. (2020) show that language models can be successfully adapted to the task domain by continuing to pretrain them in the new domain. This raises the question of whether improvements due to SOCRATIC pretraining are simply due to a better affinity of the pretraining corpus to the task domain. To answer this question, we perform continued pretraining[2] on a 1M subset of the Books3 corpus and next finetune the model on QMSum. Table 1 shows that continued pretraining slightly hurts Rouge-1 performance. In comparison, performing SOCRATIC pretraining on the same corpus improves performance by +0.41 Rouge-1. This observation rules out that improvements achieved through SOCRATIC pretraining are simply due to improved domain adaptation.

6.3 Comparing to Pre-Finetuning

Transferring information from related tasks is another approach to adapt generic models to specific tasks (Aghajanyan et al., 2021). We show in Table 1 that SOCRATIC pretraining outperforms even the best pre-finetuned BART SegEnc model, which uses additional supervision from the WikiSum dataset (Liu et al., 2018). This transfer dataset was selected from a wide range of relevant summarization datasets tested by Vig et al. (2022). Crucially, we note that transfer learning, like pre-finetuning, is orthogonal to our line of work which operates on the pretraining side. We believe that SOCRATIC can therefore be used in combination with pre-finetuning to further boost performance.

[2]For consistency, we use Fairseq to pretrain BART-large

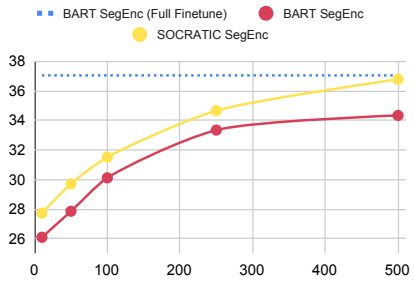


Figure 7: Few-shot performance on QMSum test set.

6.4 General vs. Specific Summaries

Both QMSum and Squality datasets contain a substantial portion of general summaries (12.5-20%) that aim to summarize the entire document in addition to those answering more specific queries. We find that our approach improves in both cases (+0.98 and +0.28 ROUGE-1 on QMSum in general and specific queries respectively). This shows that SOCRATIC pretraining improves models intended to perform a combination of general-purpose and query-focused summarization. In addition, with users increasingly interacting with language models through prompts to perform different tasks, the query-focused datasets we evaluate on become realistic testbeds for NLP systems that aim to perform well across tasks.

6.5 Few-Shot Finetuning

To show that SOCRATIC pretraining alleviates the need for labeled downstream task data, we study the few-shot learning performance of SOCRATIC and BART SegEnc models. We perform one finetuning run for each model on each subset of the task data. In Figure 7, we show that with half the QMSum examples, SOCRATIC SegEnc achieves the same performance as finetuning BART SegEnc on all of QMSum. We believe that bringing SOCRATIC pretraining closer to the downstream task of query-focused summarization lets the models learn from fewer downstream task examples.

7 Finegrained Planning Results

In this section, we evaluate the effect of SOCRATIC pretraining on the adherence to user-provided finegrained control sequences. In these experiments, the same SOCRATIC pretrained model is finetuned on task-specific data with various control strategies.

7.1 Going Beyond High-Level Questions

The queries found in QMSum and SQuALITY are only one format to encode user intent. Previous research explored other control strategies like keywords (He et al., 2020), entity chains (Narayan et al., 2021), or factoid question-answer pairs (Narayan et al., 2022). As seen in Figure 8, these strategies offer a more finegrained level of control over the summaries as they operate at the sentence level. Reference control sequences are not available for QMSum and SQuALITY so we *generate them automatically* from reference summaries. In the summarization literature, such control sequences are often modeled as intermediate plans generated before the summaries (Narayan et al., 2022; He et al., 2020). In these cases, given the input X, the model first generates the detailed plan for the summary B from $P(B|X)$, then generates the summary Y conditioning on the plan and the input x from $P(Y|B, X)$. Even if the plan B is initially generated by the model, a user can control the summary by altering the plan. In practice, we experiment with three different planning strategies.

- *Content questions*. For each sentence in the reference summary, we generate a question using the MixQG system while giving the full summary as context. These are similar to the questions that we use in our SOCRATIC pretraining. The sentence-level questions are then concatenated into a single plan for the summary. To our knowledge, we are the first to propose using content questions as finegrained plans for summaries.

- *QA blueprint*. We reimplement the recently proposed text plan in the form of a sequence of question-answer (QA) pairs (Narayan et al., 2022). First, all noun phrase answers are extracted from the reference. Then, a QG system generates questions answered by each noun phrase. The QA pairs are then filtered using round-trip consistency, rheme, and coverage criteria. The final plan consists of the concatenation of the remaining QA pairs.

- *Keywords*. We use keywords extracted from each sentence of the reference summary. We take the noun-phrase answers from the QA blueprint as keywords and concatenate them with sentence separators into a plan.

Control Strategy	Model	Summary Rouge1	Summary Rouge2	Summary RougeL	Summary BS-R	Control Plan Rouge1	Control Plan Rouge2	Control Plan RougeL	Control Plan Leven. Edit
Content Questions	BART-Large SegEnc	35.3	11.6	30.7	86.95	42.3	**23.4**	41.6	0.77
	+ PEGASUS Pret.	35.4	11.8	30.9	87.03	41.7	22.9	41.0	0.74
	+ SOCRATIC Pret.	**36.0**	**12.1**	**31.5**	**87.15**	**42.4**	23.2	**41.7**	0.77
Blueprint QA	BART-Large SegEnc	33.5	9.3	29.4	86.62	40.2	15.7	39.2	0.85
	+ SOCRATIC Pret.	**35.4**	**10.0**	**30.6**	**86.89**	**40.7**	**15.9**	**39.6**	0.85
Keywords	BART-Large SegEnc	36.2	12.8	31.4	**87.01**	24.1	9.2	21.3	0.88
	+ SOCRATIC Pret.	**36.9**	**13.2**	**32.1**	**87.01**	**25.0**	**10.0**	**22.1**	0.88

Table 2: Results on different control strategies on QMSum (results averaged over five random seeds).

Original Text: In a group discussion about a philosophical concept, Sarah used the Socratic method by asking and answering questions to stimulate critical thinking and clarify underlying assumptions. The method helped her and her classmates achieve a deeper understanding of the concept and address disagreements. Sarah looked forward to continuing to use it in her studies.

Content Questions (Ours): How did Sarah use the Socratic method? What were the benefits of the Socratic method? What did Sarah think of the method?

Keywords: Group discussion | Sarah | Socratic method | questions | thinking | assumptions || method | classmates | understanding | disagreement || studies

Blueprint QA: What type of discussion did Sarah have about a philosophical concept? Group discussion | Who used the Socratic method? Sarah | What method did Sarah use to stimulate critical thinking? Socratic method | What did Sarah ask in the Socratic method? questions | What did Sarah clarify in the Socratic method? assumptions ...

Figure 8: Comparison of finegrained control strategies.

Oracle Strategy	Model	R-1	R-2	R-L	BS-R
Content Questions	BART-Large SegEnc	43.7	18.0	39.0	88.32
	+ SOCRATIC Pret.	**46.8**	**20.3**	**41.7**	**88.92**
Blueprint QA	BART-Large SegEnc	52.9	24.1	46.8	89.63
	+ SOCRATIC Pret.	**56.3**	**26.6**	**49.3**	**90.03**
Keywords	BART-Large SegEnc	45.7	20.2	40.5	88.73
	+ SOCRATIC Pret.	**47.5**	**21.9**	**42.5**	**89.18**

Table 3: Performance on the QMSum dataset with various oracle finegrained control strategies.

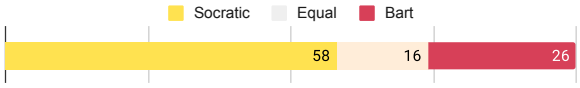


Figure 9: Annotators' finegrained planning preferences.

7.2 Comparing Control Strategies

In Table 2, we report evaluation metrics for both the model-generated summaries and plans.

We find that with all three control strategies, SOCRATIC pretraining provides a consistent improvement over the vanilla BART model and the PEGASUS pretraining on both the generated finegrained plan and summary. On the planning side, there is a small but consistent improvement, up to +0.9 Rouge-1 with keyword chain control, indicating that the model has improved planning abilities. On the summarization side, we find a more significant improvement with up to +1.9 Rouge-1 with blueprint QA control. We attribute this to a combination of improved planning and execution ability of the model from SOCRATIC pretraining.

With respect to control strategy performance, we find that our content questions obtain the highest Rouge scores (42.4 Rouge-1), outperforming keyword chains with only 25.0 Rouge-1. Despite the keyword plan having low overlap with the reference, it results in good summarization performance, so it is unclear whether the model using keyword chains learns the right correspondence between plan and summary. Moreover, the generated keyword chain would need heavier editing to obtain the reference plan compared to the content question plan (0.88 Levenstein distance compared to 0.77), making them less useful in practice.

Previous work has focused on keyword controls (He et al., 2020) and fact-oriented questions for text generation (Narayan et al., 2022), but there are inherent limitations with these approaches, which we discuss in detail in A.5.

7.3 Oracle Questions

Ideally, users can tailor generated summaries with an intervention limited to editing the generated plans. However, this requires strong adherence of generations to the finegrained plans, which we test here with oracle plans. Instead of generating both plan and summary, the system is given the oracle plans automatically extracted from the reference summaries (see 7.1). In Table 3, we observe a large improvement of +3.1 Rouge-1 over the BART SegEnc baseline. Human annotators confirm that SOCRATIC SegEnc follows oracle finegrained plans better or similarly to the baseline in 74% of instances, shown in Figure 9 and described further in A.4. This confirms our hypothesis that SOCRATIC pretraining helps ground the generations to user-provided queries. We attribute these gains to using the *Ask&Answer* mode, which introduces structure in the pretraining data by using as target text a question plan followed by its pseudo-summary answer.

We hypothesize that this structure in pretraining is what helps the model adhere to the planning step more effectively regardless of the control strategy.

8 Conclusion

In this work, we introduce SOCRATIC pretraining, a question-driven, unsupervised pretraining objective to adapt generic language models to the task of controllable summarization. SOCRATIC pretraining trains the model to generate relevant questions in a given context and then to answer them. Our experiments demonstrate the generality of our approach both on query-focused summarization and finegrained controllable summarization. We show that SOCRATIC pretraining outperforms other pretraining and prefinetuning objectives, that it cuts downstream task data requirements in half, and that it works across control strategies and domains.

9 Limitations

Downstream Tasks In this work, we focused on long-document summarization as we believe it is the task where controllable summarization is most needed. Future work could investigate the effect of SOCRATIC pretraining on other downstream applications beyond those studied here. To handle long document input we could not use the BART model with SOCRATIC pretraining adaptation directly. Instead, we applied the SegEnc architecture on top of BART. This adaptation of the pretrained model may have dampened some of the few-shot performance of SOCRATIC pretraining. We thus believe that tasks with shorter input documents for which the SegEnc architecture is not necessary would see even greater benefits in the low-resource setting.

Base Model Throughout this work, we restricted our analysis to one model architecture the SegEnc architecture with the BART base model. Previous work extensively studied the impact of different architectures for long-document query-focused summarization (Vig et al., 2022). These primarily differ in how they model long documents. The authors found SegEnc, a simple sliding window adaptation of BART, to perform best on QMSum. While the results presented here are specific to SegEnc and BART, our approach is agnostic to the underlying model architecture and is orthogonal to long-document modeling. We leave it to future work to investigate the effect SOCRATIC pretraining has on other architectures.

Evaluation Metrics As discussed in prior work (Fabbri et al., 2021b; Pagnoni et al., 2021; Gehrmann et al., 2021), there are limitations with the current automated evaluation metrics which do not strongly correlate with human judgments. Our results from these metrics should therefore be interpreted with caution and in combination with the human evaluation we performed to support them. One area in which automated metrics have been reported to perform poorly is factuality. Moreover, current factuality metrics have been designed and tested in the news domain and their performance in the out-of-domain setting (long documents and dialog data) was not systematically evaluated and is hard to interpret (Agarwal et al., 2022). In this work, we therefore choose not to report any factuality metric results.

QG Efficiency We did not optimize the efficiency of the QG component of SOCRATIC pretraining and, consequently, it is computationally expensive. Currently, given equal amounts of resources for QG and pretraining, it takes us about the same time to perform the QG phase and pretraining phase on the same amount of data. We note, however, that in low-resource scenarios, the additional compute can lead to significant benefits, as shown in our results. In addition, we did not experiment with efficient sampling strategies, and believe that improving the efficiency of the QG model inference, for example through model distillation (Hinton et al., 2015), could lead to significant efficiency gains.

Dataset Biases The datasets for pretraining and finetuning used in this work are in English and thus mainly represent the culture of the English-speaking populace. Political or gender biases may also exist in the dataset, and models trained on these datasets may propagate these biases. Additionally, the pretrained BART model carries biases from the data it was pretrained on. We did not stress test these models for biases and request that the users be aware of these potential issues in applying the models presented.

Misuse Potential and Failure Mode When properly used, the summarization models described in this paper can be time-saving. However, the current model outputs may be factually inconsistent with the input documents, and in such a case could contribute to misinformation on the internet. This issue is present among all current abstractive summarization models and is an area of active research.

References

Ester Aflalo. 2021. Students generating questions as a way of learning. *Active Learning in Higher Education*, 22(1):63–75.

Divyansh Agarwal, Alexander R. Fabbri, Simeng Han, Wojciech Kryscinski, Faisal Ladhak, Bryan Li, Kathleen McKeown, Dragomir Radev, Tianyi Zhang, and Sam Wiseman. 2022. CREATIVESUMM: Shared task on automatic summarization for creative writing. In *Proceedings of The Workshop on Automatic Summarization for Creative Writing*, pages 67–73, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Jeremy J Bornstein, Douglass R Cutting, John D Hatton, and Daniel E Rose. 1999. Interactive document summarization. US Patent 5,867,164.

Tuhin Chakrabarty, Justin Lewis, and Smaranda Muresan. 2022. Consistent: Open-ended question generation from news articles. *arXiv preprint arXiv:2210.11536*.

Richard Csaky and Gábor Recski. 2021. The Gutenberg dialogue dataset. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 138–159, Online. Association for Computational Linguistics.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog inpainting: Turning documents into dialogs. In *International Conference on Machine Learning*, pages 4558–4586. PMLR.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Angela Edmunds and Anne Morris. 2000. The problem of information overload in business organisations: a review of the literature. *International journal of information management*, 20(1):17–28.

Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021a. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021b. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2021. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv preprint*, abs/2101.00027.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang,

Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. {CTRL}sum: Towards generic controllable text summarization. *arXiv*.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *ArXiv preprint*, abs/1503.02531.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Robin Jia, Mike Lewis, and Luke Zettlemoyer. 2022. Question answering infused pre-training of general-purpose contextualized representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 711–728, Dublin, Ireland. Association for Computational Linguistics.

Wei-Jen Ko, Cutter Dalton, Mark Simmons, Eliza Fisher, Greg Durrett, and Junyi Jessy Li. 2021. Discourse comprehension: A question answering framework to represent sentence connections. *arXiv preprint arXiv:2111.00701*.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *ArXiv preprint*, abs/2010.12694.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554.

Anton Leuski, Chin-Yew Lin, and Eduard Hovy. 2003. iNeATS: Interactive multi-document summarization. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 125–128, Sapporo, Japan. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yixin Liu, Alexander R Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, et al. 2022. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. *arXiv preprint arXiv:2212.07981*.

Lidiya Murakhovs'ka, Chien-Sheng Wu, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. 2022. MixQG: Neural question generation with mixed answer types. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1486–1497, Seattle, United States. Association for Computational Linguistics.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Dipanjan Das, and Mirella Lapata. 2022. Conditional generation with a question-answering blueprint. *ArXiv preprint*, abs/2207.00397.

Shashi Narayan, Gonçalo Simoes, Ji Ma, Hannah Craighead, and Ryan Mcdonald. 2020. Qurious: Question generation pretraining for text generation. *ArXiv preprint*, abs/2004.11026.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Barak Rosenshine, Carla Meister, and Saul Chapman. 1996. Teaching students to generate questions: A review of the intervention studies. *Review of Educational Research*, 66(2):181–221.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *ArXiv preprint*, abs/2205.05131.

Jesse Vig, Alexander Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. 2022. Exploring neural models for query-focused summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1455–1468, Seattle, United States. Association for Computational Linguistics.

David Wan and Mohit Bansal. 2022. FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028, Seattle, United States. Association for Computational Linguistics.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R Bowman. 2022. Squality: Building a long-document summarization dataset the hard way. *ArXiv preprint*, abs/2205.11465.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Wenhan Xiong, Anchit Gupta, Shubham Toshniwal, Yashar Mehdad, and Wen-tau Yih. 2022. Adapting pretrained text-to-text models for long text sequences. *ArXiv preprint*, abs/2209.10052.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

A Appendix

A.1 Dataset Information

We use the QMSum and SQuALITY datasets according to their intended research purposes.

Dataset	Domain	# Ex.	Doc. Len	Sum. Len
CNN/DM	news	311K	804	60
XSum	news	226K	438	24
QMSum	meetings	1,808	9,067	70
SQuALITY	stories	625	5,200	237

Table 4: Statistics of general summarization vs. QFS datasets, length in words (Wang et al., 2022).

A.2 Training Details

We describe here the training details for SOCRATIC pretraining as well as downstream task finetuning. Our experiments rely on the Huggingface Transformers library (Wolf et al., 2020). Our code includes sample pretraining and finetuning scripts to facilitate the reproduction of our results. We use 8 Nvidia A100 GPUs to run the experiments described in this paper. We will release our code under BSD 3-Clause license.

A.2.1 Pretraining

Data Preprocessing In the Books3 corpus, documents are longer than the desired input and target texts, we therefore segment the documents to obtain roughly the desired lengths. In the UnDial dataset, the opposite is true and therefore we concatenate dialogues to obtain the desired lengths. Following this segmentation or concatenation, we mask the input text and construct the target as described in section 3 depending on the desired mode. We then truncate the input and target texts to 256 and 512 tokens respectively.

Special Tokens We introduce mode tokens and a new separator tokens to the tokenizer of the BART-large model before the pretraining adaptation step.

Training Hyperparameters We train the BART-large model for 100k steps with batch size 512, checkpointing every 10k steps. For the ablations, we use batch size of 64 and the same number of steps. In all our experiments, we use AdamW optimizer with 5k warmup steps, learning rate 3e-5, weight decay of 0.01, max grad norm of 0.1, and bfloat16. Our choice of hyperparameters is based on best practices from previous work performing pretraining adaptations of BART-large (Xiao et al., 2022; Wan and Bansal, 2022). We also performed grid-search on the learning rate on the small-scale pretraining dataset testing the values {3e-6, 3e-5, 1e-4} but finding the initial value to perform best. We use the same hyperparameters on all three pretraining corpora in our ablations.

Checkpoint validation We evaluate the checkpoints on the validation dataset of the target downstream tasks and pick the best performing checkpoint.

A.2.2 Finetuning

SegEnc Implementation We use the SegEnc implementation from the original authors. Instead of using vanilla BART-large to initialize the SegEnc model, we use one of our pretrainined models.

Finetuning Hyperparameters We use the same hyperparameters for both QMSum and SQuALITY datasets and for QFS and finegrained planning experiments. We train the SegEnc model for 10 epochs with batch size 1 and bfloat16. We use the AdamW optimizer with learning rate 5e-6. We tested the following learning rate values {5e-7, 5e-6, 5e-5, 5e-4}. We use beam search decoding with beam size of 4. Our hyperparameters follow the best performing hyperparameters found by the original authors of the SegEnc model (Vig et al., 2022). Annotations will be made available ensuring the identity of the workers remains anonymous. We will only report the answers to the questions for each example and anonymize the worker ID.

Mode While the SOCRATIC pretraining consists of both *Reconstruct* and *Ask&Answer* modes, we found that the latter performed best on the downstream tasks.

A.3 Automated Evaluation Details

We perform an automated evaluation using Rouge and BERTScore metrics following best practices from previous work. Specifically, we use the evaluation setup from Vig et al. (2022) for QMSum and the evaluation setup from Wang et al. (2022) for SQuALITY. More details and the relevant scripts can be found the in the supporting code supporting their papers. We also provide scripts to reproduce our evaluation. For BERTScore, we report recall following recommendations from Liu et al. (2022).

A.4 Human Evaluation Details

We perform a human evaluation study to confirm that variations between models are perceptible and

meaningful to human users. The study separately assesses the QFS and the finegrained planning models finetuned on the QMSum dataset. In both cases, we use 100 of the 281 examples from the QMSum test set, and three independent annotators from the Amazon Mechanical Turk platform. We restrict the study to the specific questions of the QMSum dataset as these also provide relevant text spans in the original dialogue.

We measure inter-annotator agreement with Fleiss Kappa κ (Fleiss, 1971) and obtain fair to moderate agreement in our tasks. Other studies that also rely on untrained crowd-sourced workers report similar, or sometimes even lower, agreement (Goyal et al., 2022).

QFS Task In this task, we compare the SegEnc model with SOCRATIC pretraining to Pegasus and BART pretraining. We ask annotators to select the best answer to the given query between two candidate summaries or mark if they are equally good. We provide both the reference summary and the relevant text span as supporting information. Annotator agreement on this task is $\kappa = 0.33$. The results are summarized in Figure 6 and the annotation instructions can be found in Figure 10.

Finegrained Planning Task In this task, we compare the SOCRATIC SegEnc model to the baseline BART SegEnc model in terms of their adherence to a finegrained plan. Both models are finetuned to the finegrained planning task on QMSum with the *content question* control strategy. Here we test how well they follow oracle plans automatically generated from the reference summary. The task is structured in two parts. First, for each question of the oracle plan, we ask annotators whether a sentence of the summary answers the question. We repeat for both SOCRATIC and BART summaries. On this task, we obtain moderate agreement of $\kappa = 0.49$. Next, we ask the annotators to select the best summary between the two candidates in terms of how closely it follows the plan. For the second task, the agreement is $\kappa = 0.34$. The results are summarized in Figure 9 and the annotation instructions can be found in Figure 11.

Worker Selection and Considerations An ethics review board did not review this particular protocol, but we followed prior protocols and internally-approved guidelines, such as carefully calibrating the time/HIT to ensure a pay-rate of $12/hour and letting workers know that their annotations will be used as part of a research project to evaluate the performance of summarization systems.

We selected workers according to the following criteria: HIT approval rate greater than or equal to 98%, number of HITs approved greater than or equal to 10000, and located in either the United Kingdom or the United States. The workers also passed a qualification test for a related summarization task from a prior project, ensuring that the annotators were familiar with the task of judging model-generated summaries.

A.5 Comparing Control Strategies

Using content questions for QG augmentation in SOCRATIC pretraining improves performance across control strategies, including on non-question-based finegrained controls like keyword chains (see Table 2). While most previous work has focused on keyword controls (He et al., 2020) and fact-oriented questions for text generation (Narayan et al., 2022), there are inherent limitations with these approaches. We identify important qualitative properties of queries for controllable generation below that informed our choice of content questions for SOCRATIC pretraining.

Natural To facilitate the use of controllable summarization, one overarching objective is to make the user interaction with the system as natural as possible. When evaluating how "natural" a query strategy is, we consider whether such a strategy is used by humans when they interact with one another. According to this perspective, using keywords is an unnatural query strategy. Users generally express themselves through natural language, and when inquiring about information, they use questions. Our query systems in controllable summarization should strive to reflect this and support natural queries from the users.

Unambiguous To ensure that summaries contain the intended information, it is necessary that queries refer with minimal ambiguity to the information of interest in the document. When dealing with long documents, where the same entities occur repeatedly, keywords often imprecisely describe the intended query. But it is precisely with such long documents that query-focused summarization is particularly useful. In Table 5, we show that different keyword queries about the same document have a lexical overlap of 46% of words on average and 100% in the worst-case scenario in QMSum. In

Control Type	Length	Lexical Overlap With Summ.	Lexical Overlap Across Queries	
	% of summ. len	Rouge 1	Avg. Overlap	Max. Overlap
Keywords	25%	37.9	43%	100%
Bleuprint QA	149%	65.9	22%	44%
Content Questions (ours)	48%	38.1	36%	67%

Table 5: Properties of finegrained control strategies for the QMSum dataset. We measure lexical overall between the control sequence and the reference summary. We also calculate the average and maximum lexical overlap of two control sequences from the same QMSum document but answering two different high-level queries.

comparison, content questions have a word overlap of 36% on average and no more than 67%. When formulating queries in natural language, they more richly encode the entities and their relations making them less ambiguous.

Concise Fact-oriented question-answer pairs (blueprint QA) (Narayan et al., 2022) tend to be less ambiguous than keywords (with the least lexical overlap across the three query strategies) but often end up requiring more text than the summary itself. On average, blueprint QA uses 50% more words than the summary (see Table 5). This makes this query strategy impractical for controllable summarization where the concision of the query is a desirable property.

Instructions

In this task, you will compare two candidate summaries and pick the one that provides the most *informative and correct* answer to the given question.
To correctly solve this task, follow these steps:

1. Carefully read the question, reference summary, and candidate summaries. If needed refer to the dialogue.
2. Compare the two candidate summaries according to the following criteria:
    a. *Informativeness of the summary.* How much information does the answer contain? The more informative the better.
    b. *Relevance of the information.* How relevant to the question is the information in the answer? The more relevant the information the better.
    c. *Correctness of the information.* How correct is the information in the answer? The fewer errors in the answer the better.
3. **Pick the summary that provides the most informative and correct answer to the given question.**

Note that the reference summary is provided as a guide for a satisfactory answer and that candidate summaries might be more informative than the reference.

Examples

Reference: Product Manager said that it is important to think of the weight of the device. User Interface agreed and mentioned that the current design was too heavy compared to the competition. Product manager concluded that reducing the weight of the device should be the focus of the team's work for the following week.

Example 1

Question: What what said about the weight of the device?
Summary 1: Product manager said weight is important. User Interface agreed the team should focus on reducing the device's weight.
Summary 2: Product manager noted that the device felt heavy and that the team might have overlooked this aspect. User interface mentioned the current design was heavier than the competition. Product manager concluded that the team should focus on a lighter design and present it during next week's meeting.
Explanation: Summary 2 is correct. Summary 2 is more informative than Summary 1. Summary 1 also contains a mistake: It was Product Manager that told the team to focus on reducing the weight of the device, not User Interface.

Warning: Annotations will be checked for quality against control labels, **low quality work will be rejected.**

Figure 10: QFS human annotation instructions.

Instructions

Part 1

In this task, you will be given a list of questions and a summary. For each question, you will **determine whether it is correctly answered by one of the sentences of the summary**. Each sentence of the summary can only correspond to **one** specific question. **If the question says BLANK, please mark "Not Answered."**
You will judge two summaries in this fashon.

Part 2

After judging two summaries in Part 1, you will **pick the summary that most closely answers the list of questions**. You will pick the best summary based on the number of questions it correctly answers and if the answers appear in the same order as the questions. We do provide an option to rate the summaries as equal, but note that **this is very rare/may not occur**.

Example

Part 1

Task: Which one of the specific questions is answered by a sentence of the summary?
Summary 1: Product manager noted that the device felt heavy and that the team might have overlooked this aspect. User interface mentioned the current design was heavier than the competition. Product manager concluded that the team should focus on a lighter design and present it during next week's meeting.
Questions:

- Question 1: What did Product manager note about the device?
- Question 2: What did User interface mention about the competition?
- Question 3: What was the conclusion?

Explanation:
Each one of the three questions is answered by a different sentence of summary 1.

Task: Which one of the specific questions is answered by a sentence of the summary?
Summary 2: Product manager noted that the device felt heavy and that the team might have overlooked this aspect. Product manager concluded that the team should focus on a lighter design and present it during next week's meeting.
Questions:

- Question 1: What did Product manager note about the device?
- Question 2: What did User interface mention about the competition?
- Question 3: What was the conclusion?

Explanation:
Question 1 and 3 are answered by summary 2. Question 2 was not answered by any sentence in the summary 2.

Part 2

Task: Which summary better answers the following questions in their order?
Summary 1: Product manager noted that the device felt heavy and that the team might have overlooked this aspect. User interface mentioned the current design was heavier than the competition. Product manager concluded that the team should focus on a lighter design and present it during next week's meeting.
Summary 2: Product manager noted that the device felt heavy and that the team might have overlooked this aspect. Product manager concluded that the team should focus on a lighter design and present it during next week's meeting.
Questions:

- Question 1: What did Product manager note about the device?
- Question 2: What did User interface mention about the competition?
- Question 3: What was the conclusion?

Explanation:
Summary 1. Summary 1 answers 3/3 of the questions while Summary 2 only 2/3.

Warning: Annotations will be checked for quality against control labels, **low quality work will be rejected.**

Figure 11: Finegrained planning human annotation instructions.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 9

☑ A2. Did you discuss any potential risks of your work?
Section 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 8

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We use scientific artifacts in sections 3 to 6

☑ B1. Did you cite the creators of artifacts you used?
We cite the artifacts as they are introduced in the paper in sections 2 to 6 and in the appendix.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Appendix

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4 and Appendix

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4 and Appendix.

C ☑ Did you run computational experiments?

Section 4 to 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Section 4 and the Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sections 4, 5 and 6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4 and Appendix

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Sections 5,6 and appendix

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Appendix

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Appendix

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
An ethics review board did not review this particular protocol, but we followed prior protocols and internally-approved guidelines, such as carefully calibrating the time/HIT to ensure a pay-rate of $12/hour and letting workers know that their annotations will be used as part of a research project to evaluate the performance of summarization systems.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix